Description

MULTIMEDIA DATA REPRODUCING APPARATUS, AUDIO DATA RECEIVING METHOD AND AUDIO DATA STRUCTURE THEREIN

Technical Field

[1] The present invention relates to audio data transmission, and more particularly, to a multimedia data reproducing apparatus, a method of receiving audio data using a hyper text transport protocol (HTTP) and an audio data structure used for the apparatus and method.

Background Art

[2] FIG. 1 illustrates a process of requesting an audio file from a server and receiving the requested file by a terminal receiving data over the Internet.

[3] Referring to FIG. 1, web browser software, such as Internet Explorer, is installed on a terminal 110 receiving data over the Internet. The terminal 110 can request web data stored on a server 120 to be transmitted using a predetermined protocol via the web browser software.

[4] When the terminal 110 requests an audio.ac3 file, which is a kind of compressed audio file, the terminal 110 transmits a file request message 130 to the server 120. The server 120 transmits a response message 140 to the terminal 110 and then transmits audio data to the terminal 110.

[5] Here, a generally used protocol is an HTTP protocol. The received audio data is temporarily stored in a buffer memory included in the terminal 110, decoded by a decoder reproducing data, and output as analog audio.

[6] In detail, markup resource data includes HTML files, image files, script files, audio files, and video files. The terminal 110, which receives the markup resource data, is connected to a web server, on which the markup resource data is stored, using the HTTP protocol. For example, if a user wants the terminal 110 to access a site www.company.com and download an audio.ac3 file, the terminal 110 executes the browser and accesses the server 120 by typing in 'http://www.company.com' in URL (Uniform Resource Location) field. After accessing the server 120, the file request message 130 is transmitted to the server 120. The server 120 transmits the response message 140 to the terminal 110.

[7] The server provides the stored markup resource data. Since the terminal 110 requests the audio.ac3 file, the server 120 transmits the audio.ac3 file to the terminal

110. The terminal 110 stores the received audio.ac3 file in the buffer memory. The decoder included in the terminal 110 decodes the audio.ac3 file stored in the buffer memory and outputs the decoded file as analog audio.

[8] In a conventional method of transmitting markup resource data, the terminal 110 requests a complete file and the server 120 transmits the complete file, or when a large file, such as audio data, is transmitted, the terminal 110 requests the file by defining in advance a range to be transmitted and the server 120 transmits a portion of the file corresponding to the range.

[9] However, when data is encoded temporally, and when data to be transmitted is defined according to a time at which it is to be transmitted, as in audio data, it is difficult to use the conventional method. For example, if various kinds of audio files, such as MP3, MP2, and AC3, exist, when the same time information of the audio files is transmitted to the server 120, and when audio data corresponding to the time information is requested, it is difficult to use the conventional method since locations of files corresponding to the time information are different for each kind of audio file.

Disclosure of Invention

Technical Solution

[10] The present invention provides a method of receiving audio data using an HTTP protocol, not a complex audio/video streaming protocol, a structure of received audio meta data, and a structure of audio data.

[11] The present invention also provides a multimedia data reproducing apparatus capable of reproducing audio data in synchronization with audio data and video stored in a DVD.

Advantageous Effects

[12] As described above, according to embodiments of the present invention, audio data is received using an HTTP protocol, not a complex audio/video streaming protocol, and output in synchronization with video data.

[13] For example, a DVD includes movie contents and video in which a director explains producing procedures of the movie (director's cut). The explanation is mostly produced in one language. Accordingly, a film producing company must produce a special DVD to provide Korean content. Therefore, since only audio produced with various languages is downloaded over the Internet and output in synchronization with original DVD video, problems of producing a special DVD can be overcome.

Description of Drawings

[14] FIG. 1 illustrates a process of requesting an audio file from a server and receiving

the requested file by a terminal receiving data over the Internet;

[15] FIG. 2 is a block diagram of a terminal;

[16] FIG. 3 is a block diagram of a server;

[17] FIG. 4 illustrates a process by which a terminal receives audio data from a server using meta data;

[18] FIG. 5 is a table showing request messages and response messages used to communicate between a terminal and a server;

[19] FIG. 6 illustrates the configuration of an audio.ac3 file;

[20] FIG. 7 is a block diagram of a terminal including a round type buffer;

[21] FIGS. 8A and 8B are detailed diagrams of chunk headers according to embodiments of the present invention;

[22] FIG. 9 illustrates a process of reading chunk audio data stored in a buffer, decoding the chunk audio data, synchronizing the decoded chunk audio data with video data, and outputting the synchronized audio and video data; and

[23] FIG. 10 is a flowchart illustrating a method of calculating an initial position of audio data according to an embodiment of the present invention.

Best Mode

[24] According to an aspect of the present invention, there is provided a multimedia data reproducing apparatus comprising: a decoder receiving AV data, decoding the AV data, and reproducing the AV data in synchronization with predetermined markup data related to the AV data; and a markup resource decoder receiving location information of video data being reproduced by the decoder, calculating a reproducing location of the markup data related to the video, and transmitting the reproducing location of the markup data to the decoder.

[25] According to another aspect of the present invention, there is provided a method of receiving audio data, the method comprising: receiving meta data including attribute information of audio data from a server; calculating initial position information of the audio data, transmission of which is requested, according to the attribute information included in the meta data; and transmitting the calculated initial position information to the server and receiving the audio data corresponding to the initial position.

[26] According to another aspect of the present invention, there is provided a method of calculating a location of audio data, the method comprising: converting initial time information of data, transmission of which is requested, into the number of frames included in the audio data; converting the number of frames into initial position information of a chunk, which is a transmission unit of the audio data; and calculating

byte position information corresponding to the initial chunk information.

[27] According to another aspect of the present invention, there is provided a recording medium having recorded thereon audio meta data comprising: information regarding a compression format of audio data; information regarding the number of bytes allocated to a single frame included in the audio data; time information allocated to the single frame; information regarding the size of chunk data, which is a transmission unit of the audio data, and information regarding the size of chunk head; and location information regarding a server in which the audio data is stored.

[28] According to another aspect of the present invention, there is provided a recording medium having recorded thereon an audio data structure of comprising: a chunk head field including synchronization information determining a reference point in time for reproducing the audio data; and an audio data field in which frames forming the audio data are stored.

[29] According to another aspect of the present invention, there is provided a computer readable medium having recorded thereon a computer readable program for performing a method of receiving audio data and a method of calculating a location of audio data.

Mode for Invention

[30] Hereinafter, the present invention will now be described more fully with reference to the accompanying drawings, in which exemplary embodiments of the invention are shown.

[31] A file request message used when a terminal requests a complete audio.ac3 file from a server is:

[32] GET /audio.ac3 HTTP/1.0

[33] Date: Fri, 20 Sep 1996 08:20:58 GMT

[34] Connection: Keep-Alive

[35] User-Agent: ENAV 1.0(Manufacturer).

[36] A response message that the server transmits to the terminal in response to the file request message is:

[37] HTTP/1.0 200

[38] Date: Fri, 20 Sep 1996 08:20:58 GMT

[39] Server: ENAV 1.0(NCSA/1.5.2)

[40] Last-modified: Fri, 20 Sep 1996 08:17:58 GMT

[41] Content-type: text/xml

[42] Content-length: 655360.

[43] A file request message used when the terminal requests a certain range of the audio.ac3 file from the server is:

[44] GET /audio.ac3HTTP/1.0

[45] Date: Fri, 20 Sep 1996 08:20:58 GMT

[46] Connection: Keep-Alive

[47] User-Agent: ENAV 1.0(Manufacturer)

[48] Range: 65536-131072.

[49] If the terminal requests data from a 65536 byte position to a 131072 byte position of the audio.ac3 file as shown above, a response message from the server is:

[50] HTTP/1.0 200

[51] Date: Fri, 20 Sep 1996 08:20:58 GMT

[52] Server: ENAV 1.0(NCSA/1.5.2)

[53] Last-modified: Fri, 20 Sep 1996 08:17:58 GMT

[54] Content-type: text/xml

[55] Content-length: 65536.

[56] FIG. 2 is a block diagram of a terminal. Referring to FIG. 2, a terminal 200 includes an MPEG data buffer 201, a markup resource buffer 202, an MPEG decoder 203, and a markup resource decoder 204. The terminal 200 can receive data from a server 210 via a network or from a recording medium 205 such as a disc.

[57] A markup resource stored in the server 210 is transmitted to the markup resource buffer 202, and decoded by the markup resource decoder 204. Video data stored in the recording medium 205 is transmitted to the MPEG data buffer 201 and decoded by the MPEG decoder 203. The decoded video and markup resource are displayed together.

[58] FIG. 3 is a block diagram of a server.

[59] A server 300 includes a data transmitter 301, an audio sync signal insertion unit 302, and a markup resource storage unit 303. The data transmitter 301 transmits data to and receives data from a plurality of terminals 310, 320, and 330. The audio sync signal insertion unit 302 inserts a sync signal for simultaneously reproducing audio and video by synchronizing the audio and video when the video is reproduced. The markup resource storage unit 303 stores markup resource data such as an audio.ac3 file.

[60] FIG. 4 illustrates a process by which a terminal receives audio data from a server using meta data.

[61] A terminal 410 transmits a request message requesting meta data (audio.acp) to a server 420 in step 401. The server 420 transmits a response message to the terminal 410 in response to the request message in step 402. Then, the server 420 transmits the

meta data to the terminal 410 in step 403.

[62] The audio meta data audio.acp file is:

[63] <media version='1.0'>

[64] <data name='format' value='audio/ac3' />

[65] <data name='byteperframe' value='120' />

[66] <data name='msperframe' value='32' />

[67] <data name='chunktype' value='1' />

[68] <data name='chunksize' value=8192' />

[69] <data name='chunkheader' value='21' />

[70] <data name='location' value='http://www.company.com/ac3/audio.ac3' />

[71] </media>.

[72] As indicated above, the audio meta data includes an audio file format, the number of bytes per frame, time for reproducing a single frame, a chunk type, the size of chunk, the size of a chunk header, and a location of stored audio data. The terminal 410 stores the received audio meta data audio.acp file in a buffer memory included in the terminal 410. Here, the audio.acp meta data can be read from a disc or received from a server via a network. The audio.acp meta data can also be transmitted as any type including a file type.

[73] The terminal 410 receives the audio.acp meta data and calculates a location of audio data to be read in step 404. A method of calculating the location of the audio data will be described later. When the location is calculated, the terminal 410 transmits a message requesting the actual audio file audio.ac3 to the server 420 in step 405. The server transmits a response message to the terminal 410 in response to the audio file request message in step 406 and then transmits audio.ac3 audio data to the terminal in step 407.

[74] FIG. 5 is a table showing request messages and response messages used to communicate between a terminal and a server.

[75] Referring to FIG. 5, messages transmitted from a terminal to a server include a meta data request message and an ac3 file request message, and messages transmitted from the server to the terminal include response messages in response to the request messages.

[76] FIG. 6 illustrates the configuration of an audio.ac3 file.

[77] An audio.ac3 file includes chunk header fields 610 and 630 and ac3 audio data fields 620 and 640. The chunk header fields 610 and 630 include synchronization information determining a temporal reference point for reproducing audio. The ac3 audio

data fields 620 and 640 include audio data including a plurality of frames. A single audio frame can be included in a single ac3 audio data field, and the single audio frame, such as a fourth frame 624, can be divided into two.

[78] A process of calculating a location of audio data that a terminal requests from a server is as follows.

[79] The terminal calculates the number of bytes corresponding to an initial position requested by the terminal by analyzing audio meta data audio.acp stored in a buffer memory included in the terminal. For example, if the initial position of a file requested by the terminal is 10 minutes 25 seconds 30 milliseconds, the terminal converts the initial position into a unit of milliseconds in advance. In this case, 10:25:30 = 625,030 milliseconds. The calculated value is converted into the number of frames using the reproducing time per frame (ms/frame) used in the audio meta data.

[80] The number of frames is calculated as 625,030/32 = 19,532, and accordingly, an audio data frame following the 19,532th frame is the initial position. Also, a chunk to which the 19,533th frame belongs is calculated. That is, the size of 19,532 frames is calculated as 19,532*(the number of bytes allocated to a frame) = 19,532*120 = 2,343,840 bytes.

[81] The size of data included in the ac3 audio data field 620, not including the chunk header field 610, is (the size of chunk - the size of chunk header) = 8,192 - 21 = 8,171. When the size of total frames is divided by the size of data, 2,343,840/8,171 = 286 chunks. Therefore, audio data starting from a 287th chunk is received. Here, a location of the 287th chunk converted into a unit of bytes is 286*(the size of chunk), a 2,342,912th byte position.

[82] The terminal transmits the following message including byte position information calculated as described above to the server to receive audio data:

[83] GET /audio.ac3 HTTP/1.0

[84] Date: Fri, 20 Sep 1996 08:20:58 GMT

[85] Connection: Keep-Alive

[86] User-Agent: ENAV 1.0(Manufacturer)

[87] Range: 2342912-2351103.

[88] The server transmits an audio data file audio.ac3 to the terminal. Here, the ac3 file can be read from a disc or received from the server via a network.

[89] FIG. 7 is a block diagram of a terminal including a round type buffer.

[90] Referring to FIG. 7, a terminal 700 stores a received markup resource data audio.ac3 file in a markup resource buffer 702 included in the terminal 700. The

markup resource buffer 702 is a round type buffer and consecutively receives and stores data in multiple chunk units. A markup resource decoder 704 decodes the audio.ac3 file stored in the round type markup resource buffer 702 and outputs the decoded audio.ac3 file.

[91] DVD AV data stored in a recording medium 705, such as a disc, is transmitted to a DVD AV data buffer 701, and a DVD AV decoder 703 decodes the DVD AV data. Finally, the DVD AV data decoded by the DVD AV decoder 703 and the audio.ac3 file decoded by the markup resource decoder 704 are reproduced simultaneously.

[92] FIGS. 8A and 8B are detailed diagrams of chunk headers according to embodiments of the present invention.

[93] A chunk header according to an embodiment of the present invention can be defined to follow the ISO/IEC-13818 Part 1 and a DVD standard such that a DVD file can be easily decoded. As shown in FIG. 8A, in a program stream (PS), the chunk header includes a pack header 810, a system header 820, and a PES header 830, which are written in ISO/IEC-13818. Also, only one of the pack header 810 and the system header 820 can be included in the chunk header. As shown in FIG. 8B, in a transport stream (TS), the chunk header includes a TS packet header 840 and a PES header 850.

[94] A presentation time stamp (PTS) of chunk data is included in the PES headers 830 and 850. If a fragmented frame exists at an initial position of an audio data field, the PTS indicates an initial position of a full frame.

[95] FIG. 9 illustrates a process of reading chunk audio data stored in a buffer, decoding the chunk audio data, synchronizing the decoded chunk audio data with video data, and outputting the synchronized audio and video data.

[96] Synchronization between chunk audio and DVD video is performed as follows.

[97] A markup resource decoder 704 confirms a reproducing time position of current DVD video. If it is assumed that the reproducing time position is 10 minutes 25 seconds 30 milliseconds as above, a location of relevant chunk audio can be easily determined. A method of reproducing audio using an ECMAScript will now be described using APIs.

[98] [obj].elapsed_Time is API transporting reproducing time position information of the DVD video.

[99] Also, regardless of whether synchronization with the DVD video is required and whether synchronization with the reproducing time position information of the DVD video is required when the chunk audio is synchronized and reproduced, the API: [obj].playAudioStream('http://www.company.com/audio.acp','10:25:30',true), designating

where the chunk audio is located is required.

[100] The above API indicates that a designated audio meta file, such as 'http://www.company.com/audio.asp', has been downloaded and decoded, and when the DVD video is being reproduced for 10 minutes 25 seconds 30 milliseconds until a relevant point in time, reproduction of the chunk audio starts by synchronizing an audio frame obtained by a PTS calculation of a chunk audio stream corresponding to the time.

[101] However, the API below is used when an audio clip is reproduced when the audio clip is reproduced as an infinite loop without synchronization or when the audio clip is reproduced only once:

[102] [obj].playAudioClip('http://www.company.com/audio.acp', -1).

[103] The API is used for downloading and decoding a designated audio meta file from 'http://www.company.com/audio.acp', downloading a relevant audio clip to a markup resource buffer 702, and reproducing the audio clip using the infinite loop.

[104] Here, instead of forming a file including the audio meta data, it is also possible to calculate the audio meta data using a program language (for example, Javascript, Java language) or a tag language (for example, SMIL, XML), directly extract information related to frames, and reproduce the audio clip.

[105] Also, embodiments of the present invention can be applied to not only audio data but also multimedia data configured with a fixed bitrate, for example, media data such as video, text and animation graphic data. That is, if the video, text and animation graphic data have a chunk data configuration, it is possible to reproduce the video, text and animation graphic data in synchronization with the DVD video.

[106] FIG. 10 is a flowchart illustrating a method of calculating an initial position of audio data according to an embodiment of the present invention.

[107] Reproduction initial time information of an audio file is converted into the number of frames forming audio data in step S1010. The number of frames is converted into an initial position of a chunk in step S1020. Byte position information corresponding to the initial position of the chunk is calculated in step S1030. The byte position information is transmitted to a server in step S1040, and the audio data, starting from the desired position, is received from the server.

[108] The invention can also be embodied as computer readable codes on a computer readable recording medium. The computer readable recording medium is any data storage device that can store data which can be thereafter read by a computer system. Examples of the computer readable recording medium include read-only memory

(ROM), random-access memory (RAM), CD-ROMs, magnetic tapes, floppy disks, optical data storage devices, and carrier waves (such as data transmission through the The Internet). The computer readable recording medium can also be distributed over network coupled computer systems so that the computer readable code is stored and executed in a distributed fashion.

[109] While this invention has been particularly shown and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that va rious changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The exemplary embodiments should be considered in descriptive sense only and not for purposes of limitation. Therefore, the scope of the invention is defined not by the detailed description of the invention but by the appended claims, and all differences within the scope will be construed as being included in the present invention.